EE623 Computer Vision
Spring 2024

Assignment 1 – 170 pts (includes 30 bonus points option)
Due: Jan 31, 11 59pm

Provide a link to your Google Colab or Kaggle notebook on Canvas

Q1 – 50 points – 5 pts each for 1.1-1.5 and 25 pts for Q1.6

1. Load MNIST dataset
2. Take one random image from each class and estimate L1 distance to every other class. Display these results as cell output
3. Load CIFAR10 dataset
4. Take one random image from each class and estimate L1 distance to every other class. Display these results as cell output
5. Find the closest neighboring class (K=1) based on L1 distance for each image class and provide your thoughts on why these images from different class looks similar.
6. Repeat Steps 1-5 but replace L1 distance with L2 distance

Q2 – 50 points – 5 pts each for 2.1-2.4 and 30 pts for Q2.5

1. Divide the MNIST data in train (80%) and test (20%)
2. Based on L1 distance function, run K-nearest neighbor classifier for K=5 on test set.
3. Estimate the computing time to complete testing
4. Repeat steps 2 and 3 for K=25, 55, and 105
5. Repeat steps 2,3, and 4 with L2 as a distance function

Q3 – 50 points – 10 pts each

1. Implement the same code in the link provided in Resources section (R3) for MNIST dataset
2. Repeat step 1 by changing the number of neurons in the second layer to same size as your input
3. Repeat step 1 by changing the number of neurons in the second layer to 2 times the size of your input
4. Repeat step 1 by adding another dense layer with (your chosen number) neurons
5. Compare the accuracy for the 3 above classifiers from 1-4

Q4 – 20 points

1. Summarize the key observations in Q1, Q2, and Q3.

Bonus Q – 30 points – 15 pts each

1. Apply KNN to fashion MNIST data (provided in the link R1.3)
2. Compare accuracies of MLP on fashion MNIST and KNN from Bonus Q1.

Resources

1. To load MNIST and CIFAR10 dataset from Tensorflow and Keras
   Module: tf.keras.datasets | TensorFlow v2.15.0.post1
2. Implementing KNN from scratch
   Implementing KNN from Scratch | Nikita Kozodoi
3. Multi-layer Perceptron for image classification
   Basic classification: Classify images of clothing | TensorFlow Core